How do I build MeshKit for EBMesh?

0. Prerequisites: If you'd like to read solid model based geometry files (iges, step, brep, sat, sab... etc.) MeshKit requires iGeom implementation, so you'll need to build/install it first. See CgmFromScratch for instructions. For iGeom build, you can choose a geometry engine such as OpenCASCADE and ACIS. OpenCASCADE is an opensource which can be downloaded from http://www.opencascade.org/getocc/download/loadocc/.

If you just read facet based geometry files (hdf5, genesis, stl ... etc.), you don't need to build iGeom implementation above.

In addition, MeshKit EBMesh requires iMesh implementation for its mesh data structure, so you need to build/install it too. See MoabFromScratch.

1. Grab the code straight from the repository using Subversion, by running 'svn co https://svn.mcs.anl.gov/repos/fathom/MeshKit/trunk MeshKit'.

2. Run 'autoreconf -fi' in the main source directory. NOTE: MeshKit relies on the autotools (autoconf, libtool, automake) that come with most LINUX/UNIX distributions. If they're not part of your OS, or you're getting mysterious autotools errors, you might want to try downloading/building more recent versions of those packages then trying again. On windows, try running cmake (though this is only indirectly supported, meaning it may not be up to date).

3. In the top-level source directory, run the configure script, with the various options you want. To get a list of options, run ./configure --help. To use EBMesh, you will need "./configure --with-imesh=(iMesh build directory path) --with-igeom=(iGeom build directory path)". You can remove "--with-igeom", if you only use facet based geometry.

4. Go to sub directory "algs/test" from the MeshKit top-level source directory. Run "make EBMesh_test" to make a EBMesh tool test executable. You can execute "EBMesh_test" with several options.

If you have problems running 'autoreconf -fi' or suspect that the build system is broken do to improperly installed versions of the GNU autotools then see: AutoToolsIssues.

When using --with-cgm option and CGM is build using --with-cubit option you might encounter segmentation fault and *** glibc detected *** errors. See this link for details.